

## On-Line Computer Graphics Notes

# SCALING TRANSFORMATIONS

Kenneth I. Joy  
Visualization and Graphics Research Group  
Department of Computer Science  
University of California, Davis

### Overview

Scaling, like translation is a simple transformation which just scales the coordinates of an object. It is specified either by working directly with the local coordinates, or by expressing the coordinates in terms of Frames

---

### Development of the Transformation via Scaled Frames

Given a frame  $\mathcal{F} = (\vec{u}, \vec{v}, \vec{w}, \mathbf{O})$ , a scaled frame would be one that is given by  $\mathcal{F}' = (a\vec{u}, b\vec{v}, c\vec{w}, \mathbf{O})$  – that is, we just expand (or contract) the lengths of the vectors defining the frame. It is fairly easy to see that we can write the frame  $\mathcal{F}'$  in terms of the frame  $\mathcal{F}$  by

$$\begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{u} \\ \vec{v} \\ \vec{w} \\ \mathbf{O} \end{bmatrix} = \begin{bmatrix} a\vec{u} \\ b\vec{v} \\ c\vec{w} \\ \mathbf{O} \end{bmatrix}$$

So a  $4 \times 4$  matrix implements a scaling transformation on frames, and any matrix of this type (for arbitrary  $a, b, c$ ) will scale the frame  $\mathcal{F}$ . We call the matrix

$$S_{a,b,c} = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

a scaling matrix and utilize matrices of this type to implement our scaling operations.

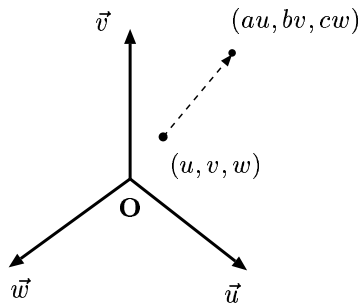
---

### Applying the Transformation Directly to the Local Coordinates

Given a frame  $\mathcal{F} = (\vec{u}, \vec{v}, \vec{w}, \mathbf{O})$  and a point  $\mathbf{P}$  that has local coordinates  $(u, v, w)$  in  $\mathcal{F}$ , if we apply the transformation  $S_{a,b,c}$  to the local coordinates of the point, we obtain

$$\begin{bmatrix} u & v & w & 1 \end{bmatrix} \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} au & bv & cw & 1 \end{bmatrix}$$

and we have scaled the point *within the frame*  $\mathcal{F}$ . An illustration of this is shown in the following figure



### Scaling about Points other than the Origin

It is difficult to see the origin of the scaling operation when working only with coordinates – so for example, consider the eight vertices of a cube centered at the origin in the Cartesian frame.

- (-1, -1, 1)
- (-1, 1, 1)
- (1, 1, 1)
- (1, -1, 1)
- (-1, -1, -1)
- (-1, 1, -1)
- (1, 1, -1)
- (1, -1, -1)

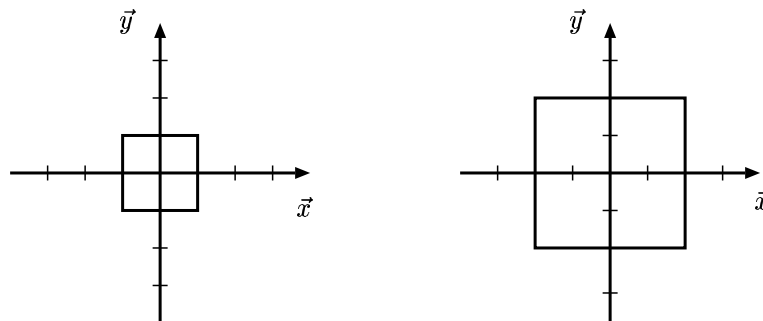
If we consider the “scaling” transformation given by

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and apply this matrix to each of the coordinates of the points relative to the standard frame, we obtain a new set of points

- $(-2, -2, 2)$
- $(-2, 2, 2)$
- $(2, 2, 2)$
- $(2, -2, 2)$
- $(-2, -2, -2)$
- $(-2, 2, -2)$
- $(2, 2, -2)$
- $(2, -2, -2)$

which is an effective scaling of the cube (The resulting cube has volume 8 times the original). This operation is illustrated in the following figure (Note that this figure is viewing the cube from along the  $z$  axis).



We note that this operation scales about the origin of the coordinate system. If the center of the object is not at the origin, this operation will move the object away from the origin of the frame. If we consider a cube with the following coordinates at its corners

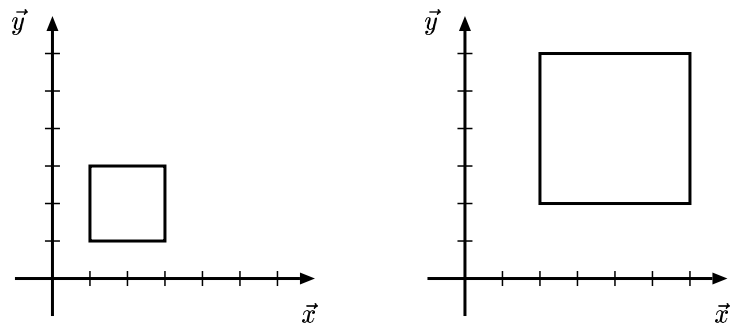
- $(1, 1, 1)$
- $(1, 3, 1)$
- $(3, 3, 1)$

(3, 1, 1)  
 (1, 1, 3)  
 (1, 3, 3)  
 (3, 3, 3)  
 (3, 1, 3)

Then by applying the above transformation, this cube is transformed to a cube with the following coordinates

(2, 2, 2)  
 (2, 6, 2)  
 (6, 6, 2)  
 (6, 2, 2)  
 (2, 2, 6)  
 (2, 6, 6)  
 (6, 6, 6)  
 (6, 2, 6)

which is a cube with volume 8 times the original, but centered at the point (4, 4, 4). This is illustrated in the following figure.



If the desired scaling point is not at the origin of the frame, we must utilize a combination of transformations to get an object to scale correctly. If the scaling point is at  $(u_s, v_s, w_s)$  in the frame, we can utilize the translation  $T_{-u_s, -v_s, -w_s}$  to first move the point to the origin of the frame, then scale the object, and finally use the translation  $T_{u_s, v_s, w_s}$  to move the point back to the origin of the scaling. These transformations are all represented by  $4 \times 4$  matrices:

$$T_{-u_s, -v_s, -w_s} S_{a,b,c} T_{u_s, v_s, w_s}$$

and we take their matrix product to create one  $4 \times 4$  matrix that gives the transformation.

---

## **Summary**

The scaling transformation can be represented by a simple  $4 \times 4$  matrix whose only entries are on the diagonal. This transformation, when applied to an object multiplies each of the local coordinates of the object by a factor – effectively scaling the object about the origin. Scaling about other points can be done by combining the scaling transformation with two translation transformations.

---

**All contents copyright (c) 1996, 1997, 1998, 1999  
Computer Science Department, University of California, Davis  
All rights reserved.**